04

Guardando para despues

Transcripción

[00:00] Hola todos y todas. Bienvenidos y bienvenidas una vez más a este curso de Git. Nos quedamos en eso de qué pasa si yo quiero guardar por ejemplo una parte de una modificación para después, o sea hice algo que no está terminado pero que sí me sirve y no quiero commitear.

[00:19] Bueno, vamos a ver. Supongamos que vamos a cambiar el curso de Ansible por Ansible: Infraestructura como código. Damos un "Ctrl + S". Pero por ejemplo yo necesito mirar si ese realmente es el nombre del curso, necesito analizarlo mejor después, pero llegó una tarea urgente en este momento, por ejemplo, me pidieron que cambie el nombre del curso de Kubernetes por ejemplo.

[00:47] Entonces necesito guardar lo que hice acá para decir: "Git mira, guardá esto en algún lugar que después yo voy a volver y voy a continuar trabajando en esto". Y ese es el nombre del concepto de Git llamado stash. Entonces vamos a la terminal de Bruno, esa de acá, ahí está. Vamos a la terminal de Bruno y hacemos un git status antes que nada.

[01:14] Vamos a ver que tenemos nuestras modificaciones y las vamos a guardar en un lugar temporal, un lugar donde puedo recuperarlas después pero que no sea un commit, ya que recuerden que no debemos subir commits que no están funcionando. Entonces voy a hacer un git stash, damos enter y vemos que se han guardado las modificaciones con un estado de work in progress.

[01:43] Eso significa ese WIP, work in progress, trabajo en proceso y nos da acá un hash. Entonces, con esto nos está indicando que se guardaron todos los cambios que no habíamos commiteado. En este caso es solamente uno. Ahora supongamos que modifico el curso de Kubernetes. Vamos a poner por ejemplo que el nombre de Kubernetes verdadero es Kubernetes: Introducción a la orquestación de containers.

[02:19] Supongamos que es eso. Damos un "Ctrl + S", entonces vamos a la terminal y voy a hacer un clear y vamos a hacer un git status, enter, vemos que está modificado en archivo, git add index.html, git commit -m y acá ponemos el mensaje: "Modificando el nombre del curso de kubernetes", enter.

[02:52] Perfecto. Ahora quiero continuar trabajando con lo que había dejado hecho en el curso de Ansible. Fíjense que los datos del curso de Kubernetes están actualizados, pero el de Ansible está en la forma anterior a que yo hubiera hecho esos cambios dentro del título.

[03:11] Entonces ahora quiero agarrar los datos guardados en el stash y traerlos a mi carpeta de trabajo. Yo tengo dos opciones. Acá dentro de la carpeta Bruno de la terminal Bruno vamos a hacer un clear. Una opción es hacer un git stash list, damos enter, y tengo una lista de todos los stash que yo puedo agarrar y colocar ahí dentro de nuestro proyecto.

[03:45] Si hago un git stash apply 0, por ejemplo, que es el número de nuestro stash que está acá, acá podríamos tener muchos. Nosotros podríamos poner el valor 0 y ahí eso lo que haría sería aplicar las modificaciones de ese stash en nuestro proyecto actualmente. Pero esas modificaciones, si utilizo este método, van a continuar dentro de la lista de stash.

[04:13] Entonces después por ejemplo tendría que hacer un git stash drop para eliminar esa modificación de la lista de stash. Pero si quisiera hacer las dos cosas al mismo tiempo, o sea, agarrar la última modificación que agregué al stash y eliminarla después de esa lista, puedo usar la segunda opción y usar el

comando git stash pop. Damos enter y este comando agarra del stash la última modificación agregada ahí en ese stash, la coloca en nuestra carpeta y después la elimina del stash.

[04:49] Entonces, por ejemplo si yo hago un git stash list, damos enter, vemos que no tengo nada. Lo que hace este stash pop es generar un merge con las modificaciones que ya teníamos y las coloca en el archivo. Fíjense por ejemplo, ahora si vamos al archivo fíjense que tengo a Ansible con las modificaciones que había hecho antes de hacer el stash, por ejemplo. Perdón, con las modificaciones que hice en el momento de hacer el stash.

[05:20] Y además de eso tengo el nuevo título de Kubernetes, que era parte del commit. Ahora por ejemplo puedo continuar editando el archivo y voy a cambiar a Ansible y supongamos que el nombre verdadero era: "Su infraestructura como código". Ahora sí estoy listo para hacer un commit, entonces vamos a hacer un "Ctrl + S", vamos a ir acá. Voy a hacer un clear.

[05:49] Y hacemos un git add index.html, damos un enter, git commit -m y acá vamos a poner: "Modificando el nombre del curso de Ansible", por ejemplo. Cerramos comillas, enter y después de haber hecho todas las modificaciones es importante enviarlas a nuestro servidor. Entonces vamos a hacer un git push servidorlocal master, enter. Bien, hemos enviado las modificaciones. Siempre es conveniente mantener actualizado nuestro repositorio remoto.

[06:30] Bueno, ahora por ejemplo vamos a hacer un git log --oneline, enter, y estamos viendo que hemos hecho muchos commits. Inclusive por ejemplo tengo commits de merge, tengo el commit este de un merge de esa branch 'lista' y si yo quisiera navegar, por ejemplo, hacer que mi código esté en el estado que estaba en el momento que hice ese merge con la branch lista.

[07:02] ¿Cómo puedo hacer eso? ¿Cómo puedo viajar en el tiempo hasta ahí y ver qué es lo que tenía nuestro código? Vamos a hablar sobre eso en el próximo video.